Global Optimization for Black-box Simulation via Sequential Intrinsic Kriging

Mehdad, E.; Kleijnen, Jack P.C.

Publication date:
2014

Document Version
Early version, also known as pre-print

No. 2014-063

GLOBAL OPTIMIZATION FOR BLACK-BOX SIMULATION VIA SEQUENTIAL INTRINSIC KRIGING

By

Ehsan Mehdad, Jack P.C. Kleijnen

16 October, 2014

ISSN 0924-7815
ISSN 2213-9532

Global Optimization for Black-box Simulation via Sequential Intrinsic Kriging

Ehsan Mehdad and Jack P.C. Kleijnen
Tilburg School of Economics and Management, Tilburg University

October 15, 2014

Abstract

In this paper we investigate global optimization for black-box simulations using metamodels to guide this optimization. As a novel metamodel we introduce intrinsic Kriging, for either deterministic or random simulation. For deterministic simulation we study the famous 'efficient global optimization' (EGO) method, substituting intrinsic Kriging for universal Kriging. For random simulation we investigate a state-of-the-art two-stage algorithm accounting for heteroscedastic variances of the simulation responses, and introduce a new variant with the following two features: (1) this variant uses intrinsic Kriging; (2) this variant uses a different procedure to allocate the total available number of replications over simulated points. We perform several numerical experiments with deterministic and random simulations, to compare (1) the classic EGO and our EGO with intrinsic Kriging; (2) the classic two-stage algorithm and our modified version. We conclude that in most experiments (1) EGO with intrinsic Kriging outperforms classic EGO; (2) there is no significant difference between the classic algorithm and our modified two-stage algorithm.

Keywords: Global optimization, Gaussian process, Kriging, intrinsic Kriging, metamodel, computer experiment, simulation

JEL: C0, C1, C9, C15, C44

1 Introduction

Optimization methods for black-box simulations—either deterministic or random—have many applications. Black-box simulation means that the input/output (I/O) function is an implicit mathematical function defined by the simulation model (computer code). In some situations, the computation of the output (response) of a single input combination may be time-consuming or 'computationally expensive'. In other situations a single simulation run is computationally inexpensive, then extremely many input combinations may be simulated; e.g., if the inputs are continuous. In both types of situation, it is common to use metamodels, which are also called emulators or surrogates. A popular method

for the optimization of deterministic simulation is *efficient global optimization* (EGO), which uses Kriging as the metamodel; see Jones et al. (1998). EGO has been adapted for random simulation with either homoscedastic variances (see Huang et al. (2006)) or heteroscedastic variances (see Picheny et al. (2013) and Quan et al. (2013)). EGO is the topic of much recent research; see Binois et al. (2014) , Couckuyt et al. (2014), and Müller and Shoemaker (2014).

Our first contribution in this paper is the use of *intrinsic Kriging* (IK) as a metamodel for the optimization of deterministic or random simulation. The idea of IK is to remove the trend from input/output (I/O)data by linear filtration of data. Unlike Kriging, IK does not require the second-order stationary condition and it may provide a more accurate fit than Kriging; see Mehdad and Kleijnen (2014).

More specifically, in deterministic simulation we use EGO with IK as the metamodel. In random simulation we use stochastic IK (SIK) combined with the two-stage sequential algorithm developed by Quan et al. (2013). This algorithm accounts for heteroscedastic noise variances and balances two source of noise; namely, spatial uncertainty due to the metamodel and random variability caused by the simulation. The latter noise is independent from one replication to another; i.e, we suppose that the streams of pseudorandom numbers do not overlap. Moreover we assume that different input combinations do not use common (pseudo)random numbers.

Our second contribution concerns the two-stage algorithm. We replace the optimal computing budget allocation (OCBA) in the allocation stage of the algorithm by an allocation rule that we build on IK. This rule allocates the additional replications over the sampled points in a way that minimizes the integrated mean squared prediction error (IMSPE).

In our numerical experiments we use test functions of different dimensionality, to study the differences between (1) EGO variants in deterministic simulation. (2) Two-stage algorithm variants in random simulation. Our major conclusion will be that in most experiments (1) EGO with IK outperform classic EGO; (2) there is no significant difference between the two-stage algorithm and our modified version.

We organize the rest of this paper as follows. Section 2 summarizes classic Kriging. Section 3 explains IK. Section 4 summarizes classic EGO, the two-stage algorithm, and our variant of this algorithm. Section 5 presents numerical experiments. Section 6 summarizes our conclusions.

2 Kriging

In this section we summarize universal Kriging (UK), following Cressie (1991, pp. 151-182). UK assumes

$$Y(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + M(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d \tag{1}$$

where $Y(\mathbf{x})$ is a random process at the point (or input combination) $\mathbf{x}$, $\mathbf{f}(\mathbf{x})$ is a vector of $p+1$ known regression functions or "trend", $\boldsymbol{\beta}$ is a vector of $p+1$

parameters, and $M(\mathbf{x})$ is a second-order stationary GP with zero mean and covariance matrix $\mathbf{\Sigma}_M$.

This $\mathbf{\Sigma}_M$ must be specified such that it makes $M(\mathbf{x})$ in (1) a second-order stationary GP; i.e., $\mathbf{\Sigma}_M$ is a function of the *distance* between the points $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ with $i, i' = 0, 1, \ldots, m$ where the subscript 0 denotes a new point and m denotes the number of old points. *Anisotropic* covariance functions use the distances along the d axes $h_{i;i';g} = |x_{i;g} - x_{i';g}|$ $(g = 1, \ldots, d)$. The most popular choice for the covariance function in $M(\mathbf{x})$ is the so-called *Gaussian* covariance function:

$$\text{cov}(\mathbf{x}_i, \mathbf{x}_{i'}) = \tau^2 \prod_{g=1}^{d} \exp\left(-\theta_g h_{i;i';g}^2\right) \text{ with } \theta_g > 0, \tag{2}$$

where τ^2 is the variance of $M(\mathbf{x})$.

Let $\mathbf{Y} = (Y(\mathbf{x}_1), \ldots, Y(\mathbf{x}_m))^\top$ denote the vector with the m values of the metamodel in (1) at the m old points. Kriging predicts Y at a (either new or old) point $\mathbf{x}_0$ *linearly* from the old I/O data $(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ is the $d \times m$ matrix with m points $\mathbf{x}_i = (x_{i;g})$ $(i = 1, \ldots, m;\ g = 1, \ldots, d)$:

$$\hat{Y}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \mathbf{Y} \text{ such that } \boldsymbol{\lambda}^\top \mathbf{F} = \mathbf{f}(\mathbf{x}_0)^\top, \tag{3}$$

where $\mathbf{F}$ is the $m \times (p+1)$ matrix with element (i, j) being $f_j(\mathbf{x}_i)$, $\mathbf{f}(\mathbf{x}_0) = (f_0(\mathbf{x}_0), \ldots, f_p(\mathbf{x}_0))^\top$, and the condition for $\boldsymbol{\lambda}$ guarantees that $\hat{Y}(\mathbf{x}_0)$ is an *unbiased* predictor. The *optimal* linear unbiased predictor minimizes the mean squared prediction error (MSPE), defined as

$$\text{MSPE}(\hat{Y}(\mathbf{x}_0)) = E(\hat{Y}(\mathbf{x}_0) - Y(\mathbf{x}_0))^2.$$

Cressie (1991, pp. 151-157) shows how to use Lagrangian multipliers to solve this constrained minimization problem, which gives the *optimal* weights and the predictor:

$$\begin{aligned} &\hat{Y}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \mathbf{Y} \\ &\boldsymbol{\lambda}^\top = \left(\mathbf{\Sigma}_M(\mathbf{x}_0, \cdot) + \mathbf{F}\left(\mathbf{F}^\top \mathbf{\Sigma}_M^{-1} \mathbf{F}\right)^{-1} \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{\Sigma}_M^{-1} \mathbf{\Sigma}_M(\mathbf{x}_0, \cdot)\right)\right)^\top \mathbf{\Sigma}_M^{-1} \end{aligned} \tag{4}$$

with $\mathbf{\Sigma}_M(\mathbf{x}_0, \cdot) = (\mathbf{\Sigma}_M(\mathbf{x}_0, \mathbf{x}_1), \ldots, \mathbf{\Sigma}_M(\mathbf{x}_0, \mathbf{x}_m))^\top$ denoting the m-dimensional vector with covariances between the outputs of the new and the m old points, and $\mathbf{\Sigma}_M$ denoting the $m \times m$ matrix with the covariances between the outputs of the old points so element (i, i') is $\mathbf{\Sigma}_M(\mathbf{x}_i, \mathbf{x}_{i'})$. The resulting minimal MSPE is

$$\begin{aligned} &\text{MSPE}(\hat{Y}(\mathbf{x}_0)) = \tau^2 - \mathbf{\Sigma}_M(\mathbf{x}_0, \cdot)^\top \mathbf{\Sigma}_M^{-1} \mathbf{\Sigma}_M(\mathbf{x}_0, \cdot) + \\ &\left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{\Sigma}_M^{-1} \mathbf{\Sigma}_M(\mathbf{x}_0, \cdot)\right)^\top \left(\mathbf{F}^\top \mathbf{\Sigma}_M \mathbf{F}\right)^{-1} \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{\Sigma}_M^{-1} \mathbf{\Sigma}_M(\mathbf{x}_0, \cdot)\right). \end{aligned} \tag{5}$$

Because the predictor is unbiased, this MSPE equals the predictor variance, which is often called the "Kriging variance".

Kriging is an exact interpolator; i.e., for the old points (4) gives a predictor that equals the observed output. For the old points the Kriging variance (5) reduces to zero.

The Kriging metamodel defined in (1) can be extended to incorporate the so-called "internal" noise in random simulation; see Opsomer et al. (1999), Ankenman et al. (2010), and Yin et al. (2011). The resulting *stochastic Kriging* (SK) metamodel at replication r of the random simulation output at $\mathbf{x}$ is

$$Z_r(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + M(\mathbf{x}) + \varepsilon_r(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d, \tag{6}$$

where $\varepsilon_1(\mathbf{x}), \varepsilon_2(\mathbf{x}), \ldots$ denotes the internal noise at input combination $\mathbf{x}$. We assume that $\varepsilon_r(\mathbf{x})$ has a Gaussian distribution with mean zero and variance $V(\mathbf{x})$ and that it is independent of $M(\mathbf{x})$.

We assumed that the external noise $M(\mathbf{x})$ and the internal noise $\varepsilon(\mathbf{x})$ in (6) are independent, so the SK predictor and its MSPE can be derived analogously to the derivation for UK in (4) and (5) except that $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_M + \boldsymbol{\Sigma}_\varepsilon$ replaces $\boldsymbol{\Sigma}_M$ where $\boldsymbol{\Sigma}_\varepsilon$ is a diagonal matrix (no common random numbers) with the variances of the internal noise $V(\mathbf{x}_i)/n_i$ on the main diagonal, and $\boldsymbol{\Sigma}_M$ still denoting the covariance matrix of Kriging without internal noise. We also replace $\mathbf{Y}$ in (4) and (5) by the sample mean $\overline{\mathbf{Z}} = (\ \overline{Z}(\mathbf{x}_1), \ldots, \overline{Z}(\mathbf{x}_m)\ )^\top$.

3 Intrinsic Kriging

In this section we explain IK, following Mehdad and Kleijnen (2014). We rewrite (1) as

$$\mathscr{Y}(\mathbf{x}) = \mathbf{F}\boldsymbol{\beta} + \mathscr{M}(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d, \tag{7}$$

where $\mathscr{Y}(\mathbf{x}) = (\mathscr{Y}(\mathbf{x}_1), \ldots, \mathscr{Y}(\mathbf{x}_m))^\top$, and $\mathscr{M}(\mathbf{x}) = (\mathscr{M}(\mathbf{x}_1), \ldots, \mathscr{M}(\mathbf{x}_m))^\top$. We no longer assume $\mathscr{M}$ is second-order stationary. Let $\mathbf{Q}$ be an $m \times m$ matrix such that $\mathbf{QF} = \mathbf{O}$ where $\mathbf{O}$ is an $m \times (p+1)$ matrix with all elements zero. Together $\mathbf{Q}$ and (7) give

$$\mathbf{Q}\mathscr{Y}(\mathbf{x}) = \mathbf{Q}\mathscr{M}(\mathbf{x}).$$

Consequently, the second-order properties of $\mathbf{Q}\mathscr{Y}(\mathbf{x})$ depend on $\mathbf{Q}\mathscr{M}(\mathbf{x})$ and *not on the regression function* $\mathbf{F}\boldsymbol{\beta}$.

To generalize the metamodel in (1), we need a stochastic process for which $\mathbf{Q}\mathscr{M}(\mathbf{x})$ is second-order stationary; such processes are called *intrinsically* stationary processes. We assume that $f_j(\mathbf{x})$ $(j = 1, \ldots, p+1)$ are mixed monomials $x_1^{i_1} \cdots x_d^{i_d}$ with $\mathbf{x} = (x_1, \ldots, x_d)^\top$ and nonnegative integers $i_1, \ldots, i_d$ such that $i_1 + \ldots + i_d \leq k$ with k a given nonnegative integer. An IRF-k is a random process $\mathscr{Y}$ for which

$$V(\mathbf{x}^*) = \sum_{i=1}^{m} \lambda_i \mathscr{Y}(\mathbf{x}_i + \mathbf{x}^*) \text{ with } \mathbf{x}_i, \mathbf{x}^* \in \mathbb{R}^d$$

is second-order stationary, and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)^\top$ is a *generalized-increment* vector of real numbers such that

$$(f_j(\mathbf{x}_1), \ldots, f_j(\mathbf{x}_m))\, \boldsymbol{\lambda} = 0 \ (j = 1, \ldots, p+1).$$

IK is based on an IRF-k. Let $\mathscr{M}(\mathbf{x})$ be an IRF-k with mean zero and *generalized covariance* matrix $\mathbf{K}$. Then the corresponding IK metamodel is:

$$\mathscr{Y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + \mathscr{M}(\mathbf{x}). \tag{8}$$

Cressie (1991, pp. 299-306) derives a linear predictor for the IRF-k metamodel defined in (8) with generalized covariance matrix $\mathbf{K}$ that is the analogue of UK. So—given the old outputs $\boldsymbol{\mathscr{Y}} = (\mathscr{Y}(\mathbf{x}_1), \ldots, \mathscr{Y}(\mathbf{x}_m))^\top$—the optimal linear predictor $\hat{\mathscr{Y}}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \boldsymbol{\mathscr{Y}}$ follows from minimizing the MSPE of the linear predictor:

$$\min_{\boldsymbol{\lambda}} E\left(\hat{\mathscr{Y}}(\mathbf{x}_0) - \mathscr{Y}(\mathbf{x}_0)\right)^2$$

IK should meet the condition $E\left(\hat{\mathscr{Y}}(\mathbf{x}_0)\right) = E\left(\mathscr{Y}(\mathbf{x}_0)\right)$, which is equivalent to

$$\boldsymbol{\lambda}^\top \mathbf{F} = \left(f_0(\mathbf{x}_0), \ldots, f_p(\mathbf{x}_0)\right). \tag{9}$$

This condition is not introduced as the unbiasedness condition but as the condition that guarantees that the coefficients of the prediction error $\lambda_1 \mathscr{Y}(\mathbf{x}_1) + \ldots + \lambda_m \mathscr{Y}(\mathbf{x}_m) - \mathscr{Y}(\mathbf{x}_0)$ create a generalized-increment vector $\boldsymbol{\lambda}_{m+1}^\top = \left(\boldsymbol{\lambda}^\top, \lambda_0\right)$ with $\lambda_0 = -1$. The IK variance, denoted by σ^2_{IK} is:

$$\sigma^2_{\text{IK}} = \text{var}(\boldsymbol{\lambda}_{m+1}^\top \boldsymbol{\mathscr{Y}}) = \sum_{i=0}^{m} \sum_{i'=0}^{m} \lambda_i \lambda_{i'} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_{i'}). \tag{10}$$

In this section we assume that $\mathbf{K}$ is known, so the optimal linear predictor is obtained through minimization of (10) subject to (9). This problem resembles the UK objective function in Section 2, with $\boldsymbol{\Sigma}_M$ now replaced by $\mathbf{K}$. Hence, the IK predictor is

$$\begin{aligned} &\hat{\mathscr{Y}}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \boldsymbol{\mathscr{Y}}, \\ &\boldsymbol{\lambda}^\top = \left(\mathbf{K}(\mathbf{x}_0, \cdot) + \mathbf{F}\left(\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F}\right)^{-1} \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)\right)\right)^\top \mathbf{K}^{-1}, \end{aligned} \tag{11}$$

where $\mathbf{K}(\mathbf{x}_0, \cdot) = \left(\mathbf{K}(\mathbf{x}_0, \mathbf{x}_1), \ldots, \mathbf{K}(\mathbf{x}_0, \mathbf{x}_m)\right)^\top$ and $\mathbf{K}$ is an $m \times m$ matrix with the (i, i') element $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_{i'})$. The resulting σ^2_{IK} is

$$\begin{aligned} &\text{MSPE}(\hat{\mathscr{Y}}(\mathbf{x}_0)) = \mathbf{K}(\mathbf{x}_0, \mathbf{x}_0) - \mathbf{K}(\mathbf{x}_0, \cdot)^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot) + \\ &\left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)\right)^\top \left(\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F}\right)^{-1} \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}(\mathbf{x}_0, \cdot)\right). \end{aligned} \tag{12}$$

IK is an exact interpolator, like UK is. Note that comparison of the predictor and MSPE of UK and IK shows that if $\mathscr{M}(\mathbf{x})$ is a second-order stationary process, UK and IK give identical results.

Obviously $\mathbf{K}$ is symmetric. Moreover $\mathbf{K}$ must be *conditionally* positive definite so

$$\text{var}(\boldsymbol{\lambda}^\top \boldsymbol{\mathscr{Y}}) = \sum_{i=1}^{m} \sum_{i'=1}^{m} \lambda_i \lambda_{i'} K(\mathbf{x}_i - \mathbf{x}_{i'}) \geq 0 \text{ such that } \left(f_j(\mathbf{x}_1), \ldots, f_j(\mathbf{x}_m)\right) \boldsymbol{\lambda} = 0,$$

where the condition must hold for $j = 1, \ldots, p+1$. Parametric models for $\mathbf{K}$ are given by Mathéron (1973).

The anisotropic version of the covariance function K for the k times integrated Brownian motion—which is an IRF-k—is:

$$K(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \prod_{g=1}^{d} \left(\theta_{0;g} + \theta_{1;g} \int_0^1 \frac{(x_g - u_g)_+^{k_g} (x'_g - u_g)_+^{k_g}}{(k_g!)^2} du_g \right), \tag{13}$$

where $\boldsymbol{\theta} = (\theta_{0;1}, \theta_{1;1}, \theta_{0;2}, \ldots, \theta_{0;d}, \theta_{1;d}) \geq 0$.

The function in (13) accepts different k for different input dimensions, so we have a vector of the orders $\mathbf{k} = (k_1, \ldots, k_d)^\top$ instead of a single scalar k for all the d input dimensions that the isotropic covariance functions have. Anisotropic covariance functions handle each input dimension separately, which makes them more flexible than isotropic covariance functions. However, this comes at the cost of estimating more parameters.

In practice, $\mathbf{K}$ is unknown so we estimate the parameters $\boldsymbol{\theta}$. For this estimation we use *restricted maximum likelihood* (REML). So we assume that $\mathscr{Y}$ in (8) is a *Gaussian* IRF-k. The REML estimator of $\boldsymbol{\theta}$ is then found through minimization of the negative log-likelihood function

$$\begin{aligned} \ell(\boldsymbol{\theta}) = (m-q)/2 \log(2\pi) - \frac{1}{2} \log \left| \mathbf{F}^\top \mathbf{F} \right| + \frac{1}{2} \log |\mathbf{K}(\boldsymbol{\theta})| + \frac{1}{2} \log \left| \mathbf{F}^\top \mathbf{K}(\boldsymbol{\theta})^{-1} \mathbf{F} \right| \\ + \frac{1}{2} \boldsymbol{\mathscr{Y}}^\top \boldsymbol{\Xi}(\boldsymbol{\theta}) \boldsymbol{\mathscr{Y}}, \end{aligned} \tag{14}$$

where $q = \operatorname{rank}(\mathbf{F})$ and $\boldsymbol{\Xi}(\boldsymbol{\theta}) = \mathbf{K}(\boldsymbol{\theta})^{-1} - \mathbf{K}(\boldsymbol{\theta})^{-1} \mathbf{F} \left(\mathbf{F}^\top \mathbf{K}(\boldsymbol{\theta})^{-1} \mathbf{F} \right)^{-1} \mathbf{F}^\top \mathbf{K}(\boldsymbol{\theta})^{-1}$. Finally, we replace $\mathbf{K}$ by $\mathbf{K}(\hat{\boldsymbol{\theta}})$ in (11) to obtain $\hat{\mathscr{Y}}(\mathbf{x}_0)$ and in (12) to obtain $\hat{\sigma}^2_{\mathrm{IK}}$.

We could require REML to estimate the optimal (integer) $\mathbf{k}^*$, but this requirement would make the optimization even more difficult; i.e., we set $\mathbf{k} = \mathbf{0}$.

Mehdad and Kleijnen (2014) also extend IK to account for simulation output that is random and has variances that change across the input space. The methodology is similar to the extension of Kriging to stochastic Kriging. We mentioned earlier that IK is an interpolator; this is not a good property for random simulation. Random simulation has sampling variability or internal noise besides the external noise that is the spatial uncertainty created by the fitted metamodel.

The extension of IK to account for internal noise with a constant variance has already been studied in the (geostatistics) literature as a so-called nugget effect. Indeed, Cressie (1991, p. 305) briefly discusses IK in case of a nugget effect, replacing $\mathbf{K}$ by $\mathbf{K} + c_0 \delta(\mathbf{h})$ where $c_0 \geq 0$, $\delta(\mathbf{h}) = 0$ if $\mathbf{h} > 0$, and $\delta(\mathbf{h}) = 1$ if $\mathbf{h} = 0$. Mehdad and Kleijnen (2014) considers the case of heteroscedastic variances.

Mehdad and Kleijnen (2014) extends the IK metamodel defined in (8) in order to incorporate the internal noise. This metamodel is similar to the one for

SK. The stochastic IK (SIK) metamodel at replication r of the random output at $\mathbf{x}$ is

$$\mathsf{Y}_r(\mathbf{x}) = \mathbf{f}(\mathbf{x})^\top \boldsymbol{\beta} + \mathsf{M}(\mathbf{x}) + \varepsilon_r(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^d, \tag{15}$$

where $\varepsilon_1(\mathbf{x}), \varepsilon_2(\mathbf{x}), \ldots$ denotes the internal noise at input combination $\mathbf{x}$. We again assume that $\varepsilon(\mathbf{x})$ has a Gaussian distribution with mean zero and variance $\mathsf{V}(\mathbf{x})$ and that $\varepsilon(\mathbf{x})$ is independent of $\mathsf{M}(\mathbf{x})$.

Our new experimental design consists of pairs $(\mathbf{x}_i, n_i),\ i = 1, \ldots, m$, where n_i is the number of replications at input combinations $\mathbf{x}_i$. The replications enable us to compute the classic unbiased estimators of the mean output and the internal variance:

$$\overline{\mathsf{Y}}(\mathbf{x}_i) = \frac{\sum_{r=1}^{n_i} \mathsf{Y}_{i;r}}{n_i} \text{ and } s^2(\mathbf{x}_i) = \frac{\sum_{r=1}^{n_i} \left(\mathsf{Y}_{i;r} - \overline{\mathsf{Y}}(\mathbf{x}_i)\right)^2}{n_i - 1}. \tag{16}$$

Because we assumed that $\mathsf{M}(\mathsf{x})$ and $\varepsilon(\mathbf{x})$ in (15) are independent, the SIK predictor and its MSPE can be derived similarly to the IK predictor and MSPE in (11) and (12)—except that $\mathbf{K}_\mathsf{M}$ will be replaced by $\mathbf{K} = \mathbf{K}_\mathsf{M} + \mathbf{K}_\varepsilon$ where $\mathbf{K}_\varepsilon$ is a diagonal matrix with the variances of the internal noise $\mathsf{V}(\mathbf{x}_i)/n_i$ on the main diagonal, and $\mathbf{K}_\mathsf{M}$ still denotes the generalized covariance matrix of IK without internal noise. We also replace $\mathscr{Y}$ in (11) and (12) by $\overline{\mathbf{Y}} = (\ \overline{\mathsf{Y}}(\mathbf{x}_1), \ldots, \overline{\mathsf{Y}}(\mathbf{x}_m)\ )^\top$. So the SIK predictor is

$$\widehat{\mathsf{Y}}(\mathbf{x}_0) = \boldsymbol{\lambda}^\top \overline{\mathbf{Y}} \text{ where}$$
$$\boldsymbol{\lambda}^\top = \left(\mathbf{K}_\mathsf{M}(\mathbf{x}_0, \cdot) + \mathbf{F}\left(\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F}\right)^{-1}\left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}_\mathsf{M}(\mathbf{x}_0, \cdot)\right)\right)^\top \mathbf{K}^{-1} \tag{17}$$

and its MSPE is

$$\text{MSPE}(\widehat{\mathsf{Y}}(\mathbf{x}_0)) = \mathbf{K}_\mathsf{M}(\mathbf{x}_0, \mathbf{x}_0) - \mathbf{K}_\mathsf{M}(\mathbf{x}_0, \cdot)^\top \mathbf{K}^{-1} \mathbf{K}_\mathsf{M}(\mathbf{x}_0, \cdot) +$$
$$\left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}_\mathsf{M}(\mathbf{x}_0, \cdot)\right)^\top \left(\mathbf{F}^\top \mathbf{K}^{-1} \mathbf{F}\right)^{-1} \left(\mathbf{f}(\mathbf{x}_0) - \mathbf{F}^\top \mathbf{K}^{-1} \mathbf{K}_\mathsf{M}(\mathbf{x}_0, \cdot)\right). \tag{18}$$

We again use REML to estimate the parameters $\boldsymbol{\theta}$ of the generalized covariance function, and replace $\mathbf{K}_\mathsf{M}$ by $\mathbf{K}_\mathsf{M}(\widehat{\boldsymbol{\theta}})$. We also need to estimate the internal noise V which is typically unknown. Let $\widehat{\mathsf{V}}(\mathbf{x}_i) = s^2(\mathbf{x}_i)$ be the estimator of $\mathsf{V}(\mathbf{x}_i)$, so we replace $\mathbf{K}_\varepsilon$ by $\widehat{\mathbf{K}}_\varepsilon = \left(\widehat{\mathsf{V}}(\mathbf{x}_1)/n_1, \ldots, \widehat{\mathsf{V}}(\mathbf{x}_m)/n_m\right)$. Finally, we replace $\mathbf{K} = \mathbf{K}_\mathsf{M} + \mathbf{K}_\varepsilon$ by $\widehat{\mathbf{K}} = \mathbf{K}_\mathsf{M}(\widehat{\boldsymbol{\theta}}) + \widehat{\mathbf{K}}_\varepsilon$ in (17) and (18). Next we explain how we choose the number of replications at each old point n_i.

We are interested in an experimental design that gives a low integrated MSPE (IMSPE). Following Mehdad and Kleijnen (2014)—who revised Ankenman et al. (2010)—we allocate N replications among m old points $\mathbf{x}_i$ such that this design minimizes the IMSPE. Let $\mathscr{X}$ be the design space. Then our goal is

$$\min_{\mathbf{n}} \text{IMSPE}(\mathbf{n}) = \int_{\mathbf{x}_0 \in \mathscr{X}} \text{MSPE}(\mathbf{x}_0, \mathbf{n}) d\mathbf{x}_0,$$

subject to $\mathbf{n}^\top \mathbf{1}_m \leq N$, and $\mathbf{n} = (n_1, \ldots, n_m)^\top$ where $n_i \in \mathbb{N}$.

Assume that N is large enough so that $\mathsf{K} \approx \mathsf{K}_\mathsf{M}$. Relaxing the integrality condition, we get the optimal allocation of the total number of replications N over m old points:

$$n_i^* \approx N \frac{\sqrt{\mathsf{V}(\mathbf{x}_i) C_i}}{\sum_{i=1}^m \sqrt{\mathsf{V}(\mathbf{x}_i) C_i}}, \text{ with } C_i = \mathbf{1}^\top \left[\mathbf{W} \circ \left(\mathbf{S}^{-1} \mathbf{J}^{(ii)} \mathbf{S}^{-1} \right) \right] \mathbf{1}, \tag{19}$$

where

$$\mathbf{S} = \begin{bmatrix} \mathbf{O} & \mathbf{F}^\top \\ \mathbf{F} & \mathsf{K} \end{bmatrix}, \ \mathbf{W} = \int \begin{bmatrix} \mathbf{f}(\mathbf{x}_0) \mathbf{f}(\mathbf{x}_0)^\top & \mathbf{f}(\mathbf{x}_0) \mathsf{K}_\mathsf{M}(\mathbf{x}_0, \cdot)^\top \\ \mathsf{K}_\mathsf{M}(\mathbf{x}_0, \cdot) \mathbf{f}(\mathbf{x}_0)^\top & \mathsf{K}_\mathsf{M}(\mathbf{x}_0, \cdot) \mathsf{K}_\mathsf{M}(\mathbf{x}_0, \cdot)^\top \end{bmatrix} d\mathbf{x}_0,$$

and $\mathbf{J}^{(ii)}$ is a $(p+1+m) \times (p+1+m)$ matrix with 1 in position $(p+1+i, p+1+i)$ and zeros elsewhere. Note that in (19) both the internal noise variance $\mathsf{V}(\mathbf{x})$ and the external noise covariance function K_M affect the allocation.

4 Global optimization

EGO is the global optimization algorithm developed by Jones et al. (1998) for deterministic simulation. It uses *expected improvement (EI)* as its criterion to balance local and global search or exploiting and exploring. The EGO steps are as follows.

1. Fit a Kriging metamodel to the old I/O data. Let $f_{\min} = \min_i Y(\mathbf{x}_i)$ be the minimum function value observed (simulated) so far.

2. Estimate $\mathbf{x}_0$ that *maximizes* $\mathrm{EI}(\mathbf{x}) = \mathbb{E}\left[\max(f_{\min} - Y_p(\mathbf{x}), 0)\right]$. Assuming $Y_p(\mathbf{x}) \sim \mathcal{N}\left(\hat{Y}(\mathbf{x}), \hat{\sigma}^2(\mathbf{x})\right)$, Jones et al. (1998) derive

$$\widehat{\mathrm{EI}}(\mathbf{x}) = \left(f_{\min} - \hat{Y}(\mathbf{x})\right) \Phi\left(\frac{f_{\min} - \hat{Y}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right) + \hat{\sigma}(\mathbf{x}) \phi\left(\frac{f_{\min} - \hat{Y}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}\right), \tag{20}$$

   where $\hat{Y}(\mathbf{x})$ is defined in (4) and $\hat{\sigma}^2(\mathbf{x})$ follows from (5) substituting estimators for τ, and $\boldsymbol{\theta}$; Φ and ϕ denote the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution.

3. Simulate the response at $\hat{\mathbf{x}}_0$ found in step 2. Fit a new Kriging metamodel to the old and new points. Return to step 1, unless the $\widehat{\mathrm{EI}}$ satisfies a given criterion; e.g., $\widehat{\mathrm{EI}}$ is less than 1% of the current best function value.

Huang et al. (2006) adapt EI for random simulation. They use the metamodel defined in (15) and assume the noise variances are identical across the

design space; i.e., $V(\mathbf{x}) = V$. They introduce the following augmented EI (AEI) to balance exploitation and exploration.

$$\widehat{\mathrm{AEI}}(\mathbf{x}) = \mathbb{E}\left[\max\left(\hat{Z}(\mathbf{x}^*) - Z_p(\mathbf{x}), 0\right)\right]\left[1 - \left(\frac{\hat{V}}{\mathrm{MSPE}(\hat{Z}(\mathbf{x})) + \hat{V}}\right)^{1/2}\right], \quad (21)$$

where $\mathbf{x}^*$ stands for the current 'effective best solution', $\mathbf{x}^* = \underset{\mathbf{x}_1,\ldots,\mathbf{x}_m}{\arg\min}[\hat{Z}(\mathbf{x}) + \mathrm{MSPE}(\hat{Z}(\mathbf{x}))]$. The second term on the right-hand side in (21) accounts for the diminishing returns of additional replications at the current best point.

Picheny et al. (2013) develop a quantile-based EI known as expected quantile improvement (EQI). This criterion lets the user specify the risk level; i.e., the higher the values for the quantile are specified, the more conservative the criterion becomes. Their algorithm accounts for a limited computation budget; moreover, to sample a new point, the algorithm also considers the noise variance at future (not yet sampled) points. However, this algorithm requires a known variance function for the noise, and it has more computational complexity than the traditional EI.

Quan et al. (2013) show that EI and AEI can not be good criteria for random simulations with heteroscedastic noise variances. They argue that an EGO-type framework for random simulation with heteroscedastic noise faces three challenges: (1) An effective procedure should locate the global optimum with a limited computing budget. (2) To keep the important balance of exploration and exploitation in random simulation, a new procedure should be able to search globally without exhaustively searching a local region; a good estimation of $f_{\min}$ is necessary especially when there are several optima close to the global optimum. (3) With a limited computing budget, it is wise to explore unexplored regions in the beginning of the search and as the budget is being expended toward the end, the focus should be on improving the current best area. Quan et al. (2013) also find no significant difference between their own two-stage algorithm and Picheny et al. (2013)'s algorithm.

Now we explain Quan et al. (2013)'s algorithm in detail. After the initial fit of a SK metamodel, each iteration of the algorithm consists of a search stage followed by an allocation stage. In the search stage, the modified expected improvement (MEI) criterion is used to select a new point. Next in the allocation stage, OCBA distributes an additional number of replications over the sampled points. The search stage locates the potential global optima, and the allocation stage reduces the noise caused by random variability at sampled points to improve the metamodel in regions where local minima exist and finally selects the global optimum.

Their algorithm contains a 'division of allocation heuristic'. The computing budget per iteration is set as a constant, but the allocation of this budget between searching and allocation stages changes as the algorithm progresses. In the beginning, most of the budget is invested in exploration (search stage). During the progress of the algorithm, the focus moves to identify the point with the lowest sample mean (allocation stage).

In the search stage, the MEI criterion is $\mathbb{E}[\max(\hat{Z}_{\min} - Z_p(\mathbf{x}), 0)]$, where $\hat{Z}_{\min}$ is the predicted response at the sampled point with the lowest sample mean, and Z_p is a normal random variable with mean $\hat{Z}(\mathbf{x})$ and estimated variance $\text{MSPE}(\hat{Z}(\mathbf{x}))$. The allocation stage addresses the random noise, so only $\text{MSPE}(\hat{Z}(\mathbf{x}))$ with estimates of $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_M$ instead of $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_M + \boldsymbol{\Sigma}_\varepsilon$ is used in the search stage. This helps the search to focus on the new points that reduce the spatial uncertainty of the metamodel. Ignoring the uncertainty caused by random variability, the MEI criterion assumes that the observations are made with infinite precision so the same point is never selected again. This helps the algorithm to quickly escape from a local optimum and brings the sampling behavior closer to the behavior of the original EI criterion and its balance between exploration and exploitation.

The allocation stage reduces random variability by allocating additional replications among sampled points. Additional replications are distributed with the goal of maximizing the probability of the correct selection (PCS) of a sampled point as the global optimum. Assume that we have m sampled points with each point $\mathbf{x}_i$ having a sample mean $\overline{Z}_i$ and sample variance $\widehat{V}(\mathbf{x}_i) = s^2(\mathbf{x}_i)$. Then the approximate probability of correct selection (APCS) can be asymptotically maximized when the available computing budget N tends to infinity and

$$\frac{n_i}{n_j} = \left(\frac{\widehat{V}(\mathbf{x}_i)/\Delta_{b,i}}{\widehat{V}(\mathbf{x}_j)/\Delta_{b,j}} \right)^2 \quad i, j = 1, \ldots, m \text{ and } i \neq j \neq b, \tag{22}$$

$$n_b = \widehat{V}(\mathbf{x}_b) \sqrt{\sum_{i=1, i \neq b}^{m} \frac{n_i^2}{\widehat{V}(\mathbf{x}_i)}}, \tag{23}$$

where n_i is the number of replications allocated to $\mathbf{x}_i$, $\mathbf{x}_b$ is the point with the lowest sample mean, and $\Delta_{b,i}$ is the difference between the lowest sample mean and the sample mean at point $\mathbf{x}_i$. Given this allocation rule, at the end of the allocation stage the sampled point with the lowest sample mean will be selected as $\hat{Z}_{\min}$.

Quan et al. (2013)'s algorithm is summarized in Algorithm 1. Before the algorithm begins, the user must specify T, B, m_0, and $r_{\min}$ where T is the total number of replications at the start, B is the number of replications available for each iteration, m_0 is the size of the initial space filling design, and $r_{\min}$ is the minimum number of replications for a new point. The size of the initial design m_0 may be set to $10d$, where d is the number of dimensions; B and $r_{\min}$ should be set such that there are sufficient replications available for the first allocation stage.

Because the starting parameters that determine the number of iterations, $I = \lceil (T - m_0 B)/B \rceil$, and the computing budget used per iteration B are set prior to collecting any data, the starting parameter settings may turn out to be unsuitable for the problem. In step 2, leave-one-out cross validation can provide feedback regarding the suitability of the initial parameters. If one or

more design points fail the cross-validation test, then the computing budget may be insufficient to deal with the noise in the response. Possible solutions include increasing B or increasing the number of design points around the point(s) that fail the cross-validation test or applying a logarithmic or inverse transformation to the response.

After successful validation, the computing budget set aside for the allocation stage $r_A(i)$ increases by a block of $\lfloor (B - r_{\min})/I \rfloor$ replications in every iteration while $r_S(i)$ decreases by the same amount for the search stage. This heuristic gives the algorithm the desirable characteristic of focusing on exploration at the start and on exploitation at the end of the search.

Algorithm 1 Quan et al. (2013)'s algorithm

Step 1. Initialization: Run a space filling design with m_0 points, with B replications allocated to each point.
Step 2. Validation: Fit a SK metamodel to the set of sample means. Use leave-one-out cross validation to check the quality of the initial SK.
Step 3. Set $i = 1$, $r_A(0) = 0$
while $i \leq I$ **do**
  $r_A(i) = r_A(i-1) + \min\left(\lfloor \frac{B - r_{\min}}{I} \rfloor, T - m_0 B - (i-1)B \right)$
  if $(T - m_0 B - (i-1)B - r_A(i)) > 0$ **then**
    $r_S(i) = B - r_A(i)$
    Step 3a. Search Stage: Sample a new point that maximizes the MEI criterion with $r_S(i)$ replications.
    Step 3b. Allocation Stage: Using OCBA (Eq. 22 and Eq. 23), allocate $r_A(i)$ replications among all sampled points.
    Step 3c. Fit a SK metamodel to the set of sample means
    $i = i + 1$
  end if
end while
The point with the lowest sample mean at the end is the global optimum.

Our modified variant of Quan et al. (2013)'s algorithm differs from Algorithm 1 as follows. (i) We use SIK instead of SK as the underlying metamodel. (ii) In the allocation stage, we use (19) instead of (22) and (23).

5 Numerical experiments

In this section we present our numerical experiments with both deterministic and random simulations. In these experiments we use a zero degree polynomial for the trend (so $p = 0$ in (1)), so UK becomes ordinary Kriging (OK). In deterministic simulation we study the performance of classic EGO versus EGO with IK. In random simulation we study the performance of Quan et al. (2013)'s algorithm versus our variant of this algorithm with SIK instead of SK and the minimum IMSE allocation rule instead of OCBA.

We tried to use the MATLAB code developed by Yin et al. (2011)—which is a building block in Quan et al. (2013)'s algorithm—to experiment with the Kriging variants (OK for deterministic simulation and SK for random simulation), but their MATLAB code crashed in experiments with $d > 1$. So we use the R package `DiceKriging` to implement OK and SK; see Roustant et al. (2012) for more details on `DiceKriging`. We implement our code for IK and SIK in MATLAB as Mehdad and Kleijnen (2014) did. In all our experiments we select $\mathbf{k} = \mathbf{0}$.

Furthermore, we select a set of $m_c = 100d$ candidate points, and as the 'winning' point we pick the candidate point that maximizes EI or MEI. For $d = 1$ we select m_0 and m_c equispaced points; for $d > 1$ we use Latin hypercube sampling (LHS) to select m_0 and m_c space-filling points, for LHS we use the MATLAB function `lhsdesign`.

As the criterion for comparing the performance of different optimization algorithms, we use the number of simulated input combinations needed to estimate the optimal input combination (say) m. As the stopping criterion we select m reaching a limit; namely, 11 for $d = 1$, 61 for the camel-back test function ($d = 2$), 65 for Hartmann-3 ($d = 3$), and 111 for Ackley-5 ($d = 5$). We select the number of starting points m_0 to be 3 for $d = 1$, 21 for $d = 2$, 30 for $d = 3$, and 51 for $d = 5$.

5.1 Deterministic simulation experiments

In this subsection we discuss our experiments with EGO for deterministic black-box test functions of different dimensionality.

We start with Gramacy and Lee (2012)'s test function with $d = 1$:

$$f(x) = \frac{\sin(10\pi x)}{2x} + (x-1)^4 \text{ and } 0.5 < x \leq 2.5. \tag{24}$$

Next we experiment with several functions that are popular in optimization; see Dixon and Szego (1978) and `http://www.sfu.ca/~ssurjano/index.html`. We experiment with: (1) Six-hump camel-back with $d = 2$ (2) Hartmann-3 with $d = 3$ (3) Ackley-5 with $d = 5$. We define the test functions with $d > 1$ in the appendix.

Figure 1 illustrates EGO with IK for the $d = 1$ test function in (24). This function has a global minimum at $x_{\text{opt}} = 0.5486$ with output $f(x_{\text{opt}})$ = -0.869; also see the curves in the left panels of the figure, where the (blue) solid curve is the true function and the (red) dotted line is the IK metamodel. We start with $m = 3$ old points, and stop after sequentially adding seven new points (shown by black circles); i.e., from top to bottom the number of points increases starting with three points and ending with 10 points. When m is 'small', the metamodel is a 'poor' approximation. The right panels of the figure displays EI as m increases.

Figure 2 displays $f_{\min}(m) = \min f(\hat{\mathbf{x}}_i)$ $(1 \leq i \leq m)$, which denotes the estimated optimal simulation output after m simulated input combinations; horizontal lines mean that the most recent simulated point does not give a

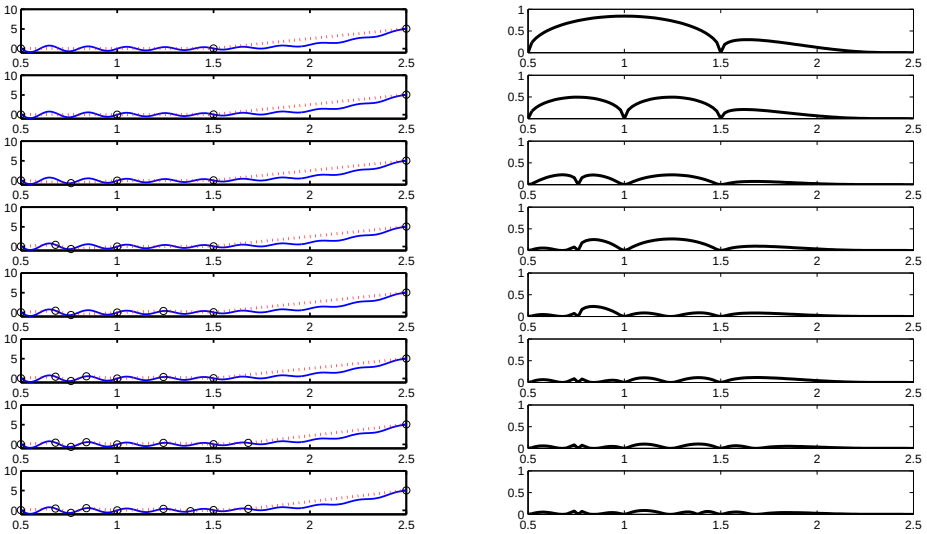
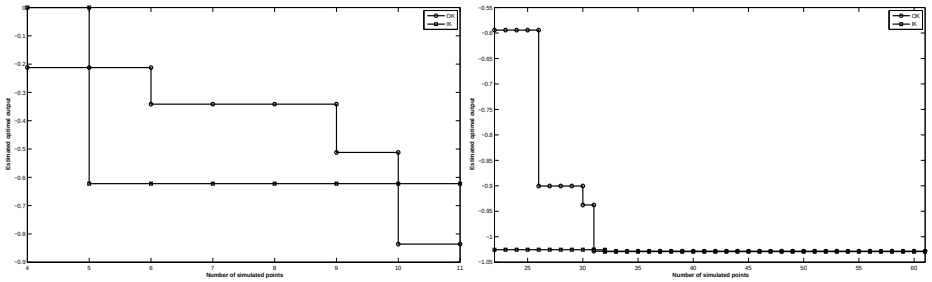
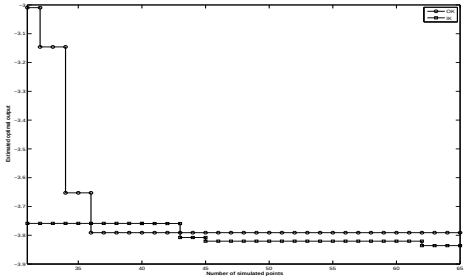


Figure 1: EGO with IK for Gramacy and Lee (2012)'s function

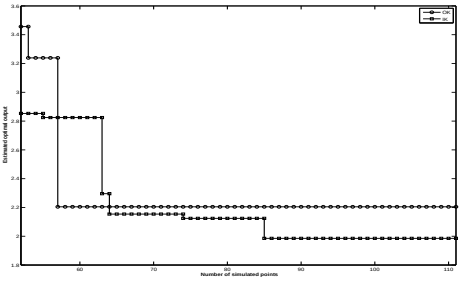


(a) EI with IK and OK in Gramacy

(b) EI with IK and OK in camel-back



(c) EI with IK and OK in Hartmann-3



(d) EI with IK and OK in Ackley-5

Figure 2: Estimated optimal output (y-axis) after m simulated input combinations (x-axis) for three test functions, for deterministic simulation

lower estimated optimal output than a preceding point. The square marker represents EGO with IK and the circle marker represents classic EGO with OK. The results show that in most experiments, EGO with IK performs better than classic EGO with OK; i.e., EGO with IK gives a better input combination after fewer simulated points.

5.2 Random simulation experiments

In this subsection we compare the performance of Quan et al. (2013)'s algorithm with our variant which uses SIK as the metamodel and a different allocation rule. In both variants we select $r_{\min} = 10$, $B = 40$ (for $d = 1$), 130 (for $d = 2$ and 3), 310 (for $d = 5$). In all test functions we augment the deterministic response with heteroscedastic noise; namely, $\mathsf{V}(\mathbf{x}_i) = (1 + |y(\mathbf{x}_i)|)^2$.

Figure 3 illustrates our variant for the $d = 1$ test function. We start with $m = 3$ old points, and stop after sequentially adding seven new points. With small m and high noise (as x increases), the metamodel turns out to be a 'poor' approximation; compare the (blue) solid curves and the (red) dashed curves. Note that in the beginning the algorithm searches the region close to the global optimum (namely, $x = 0.5486$), and after each iteration and careful allocation of added replications, the quality of the SIK fit in areas with high noise (when moving to the right) improves. The right panels of the figure displays $\widehat{\mathrm{EI}}$ as m increases.

We continue this subsection with the comparison of the two variants based on 50 macro-replications. Figure 4 displays the $f_{\min}(m) = \min_i \left(\sum_{t=1}^{50} f_t(\hat{\mathbf{x}}_i) \Big/ 50 \right)$, $1 \leq i \leq m$; which denotes the averaged (over 50 macro-replications) estimated optimal simulation output after m simulated input combinations. The square marker represents our variant and the circle marker represents Quan et al. (2013)'s original variant. The results show that the two variants are not significantly different in most sampled points except for Hartmann-3 where the original variant performs significantly better for $m = 32, \ldots, 45$ and our variant perform significantly better for $m = 31, 59, \ldots, 65$. We note that this conclusion is confirmed by paired t-tests.

6 Conclusions

We derived EGO with an IK metamodel for deterministic simulation, and we modified Quan et al. (2013)'s algorithm to have a SIK metamodel and also a new allocation rule for replications of sampled points. We numerically compared the performance through test functions of different dimensionality. The main conclusion is that in most experiments; (i) in deterministic simulations, EGO with IK performs better than Jones et al. (1998)'s EGO with OK; (ii) in random simulation, there is no significant difference between the two variants of Quan et al. (2013)'s algorithm.

In future research we may further investigate the allocation of replications in random simulation analyzed by Kriging metamodels. Furthermore, we would

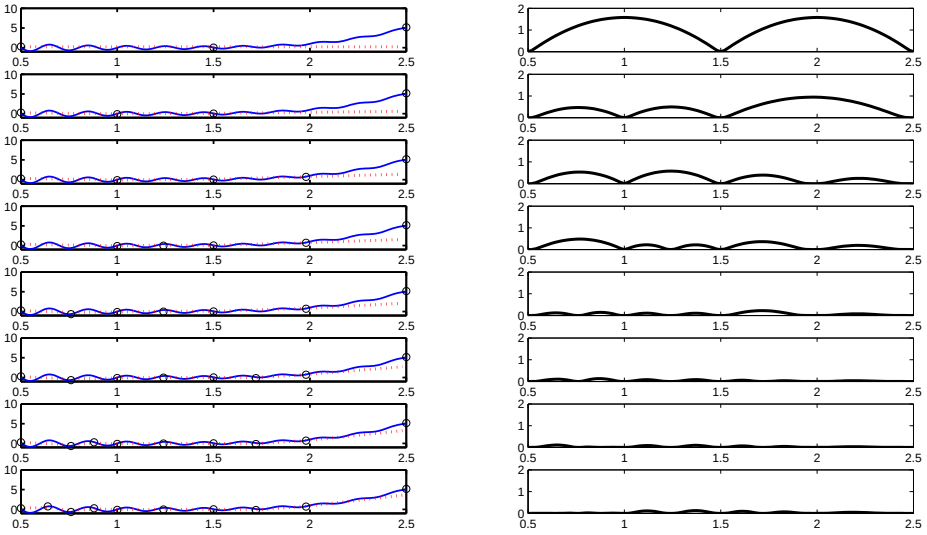
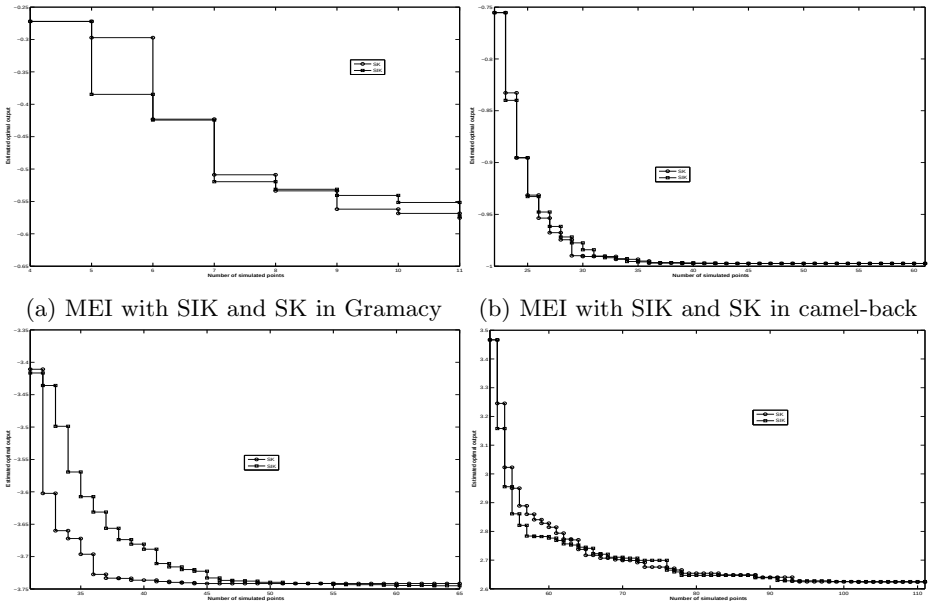


Figure 3: Our variant of Quan et al. (2013)'s algorithm for Gramacy and Lee (2012)'s function



(a) MEI with SIK and SK in Gramacy

(b) MEI with SIK and SK in camel-back

(c) MEI with SIK and SK in Hartmann-3

(d) MEI with SIK and SK in Ackley-5

Figure 4: Estimated optimal output (y-axis) after m simulated input combinations (x-axis) for three test functions, for random simulation

like to see more practical applications of our methodology.

A Test functions with $d > 1$

In this appendix we define the test functions with $d > 1$.

1. Six-hump camel-back with $-2 \leq x_1 \leq 2,\ -1 \leq x_2 \leq 1$, $\mathbf{x}_{\text{opt}}^\top = (\pm 0.0898, \mp 0.7126)$, and $f(\mathbf{x}_{\text{opt}})$ = -1.0316

$$f(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4$$

2. Hartmann-3 function with $0 \leq x_i \leq 1,\ i = 1, 2, 3$, $\mathbf{x}_{\text{opt}}^\top = (0.114614, 0.555649, 0.852547)$, and $f(\mathbf{x}_{\text{opt}})$ = -3.86278

$$f(x_1, x_2, x_3) = -\sum_{i=1}^{4} \alpha_i \exp[-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2],$$

with $\boldsymbol{\alpha} = (1.0, 1.2, 3.0, 3.2)^\top$ and A_{ij} and P_{ij} given in Table 1.

Table 1: Parameters A_{ij} and P_{ij} of the Hartmann-3 function

A_{ij}			P_{ij}		
3	10	30	0.36890	0.1170	0.26730
0.1	10	35	0.46990	0.43870	0.74700
3	10	30	0.10910	0.87320	0.55470
0.1	10	35	0.03815	0.57430	0.88280

3. Ackley-5 function with $-2 \leq x_i \leq 2,\ i = 1, \ldots, 5$, and ($\mathbf{x}_{\text{opt}} = \mathbf{0}, f(\mathbf{x}_{\text{opt}}) = 0$)

$$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{5}\sum_{i=1}^{5} x_i^2}\right) - \exp\left(\frac{1}{5}\sum_{i=1}^{5}\cos(2\pi x_i)\right) + 20 + \exp(1),$$

.

References

Ankenman, B., Nelson, B., and Staum, J. (2010). Stochastic Kriging for simulation metamodeling. *Operations Research*, 58:371–382.

Binois, M., Ginsbourger, D., and Roustant, O. (2014). Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations. *European Journal of Operational Research*. (In Press).

Couckuyt, I., Deschrijver, D., and Dhaene, T. (2014). Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, 60(3):575–594.

Cressie, N. (1991). *Statistics for spatial data*. Wiley, New York.

Dixon, L. and Szego, G., editors (1978). *Towards Global Optimisation 2*. Elsevier Science Ltd.

Gramacy, R. and Lee, H. (2012). Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22:713–722.

Huang, D., Allen, T., Notz, W., and Zeng, N. (2006). Global optimization of stochastic black-box systems via sequential Kriging meta-models. *Journal of Global Optimization*, 34:441–466.

Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492.

Mathéron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468.

Mehdad, E. and Kleijnen, J. (2014). Stochastic intrinsic Kriging for simulation metamodelling. CentER Discussion Paper No. 2014-054, Tilburg University.

Müller, J. and Shoemaker, C. (2014). Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms forcomputationally expensive black-box global optimization problems. *Journal of Global Optimization*, 60(2):123–144.

Opsomer, J., Ruppert, D., Wand, M., Holst, U., and Hossjer, O. (1999). Kriging with nonparametric variance function estimation. *Biometrics*, 55(3):704–710.

Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. (2013). Quantile-based optimization of noisy computer experiments with tunable precision (including comments). *Technometrics*, 55(1):2–36.

Quan, N., Yin, J., Ng, S., and Lee, L. (2013). Simulation optimization via Kriging: a sequential search using expected improvement with computing budget constraints. *IIE Transactions*, 45(7):763–780.

Roustant, O., Ginsbourger, D., and Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55.

Yin, J., Ng, S., and Ng, K. (2011). Kriging meta-model with modified nugget effect: an extension to heteroscedastic variance case. *Computers and Industrial Engineering*, 61(3):760–777.